APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/675,776	09/30/2003	Jimmie Earl DeWitt JR.	AUS920030481US1	6262

35525   7590   06/30/2009
IBM CORP (YA)
C/O YEE & ASSOCIATES PC
P.O. BOX 802333
DALLAS, TX 75380

EXAMINER
VU, TUAN A

ART UNIT	PAPER NUMBER
2193	

NOTIFICATION DATE	DELIVERY MODE
06/30/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ptonotifs@yeeiplaw.com

DETAILED ACTION

1.   This action is responsive to the Applicant's response filed 4/20/09.

        As indicated in Applicant's response, claims 1, 6, 26, 29-30, 32, 35, 38-39 have been

amended and claims 49-53 added. Claims 1, 6, 26, 29-30, 32, 34-35, 38-39, 49-53 are pending

in the office action.

Double Patenting

2.   The nonstatutory double patenting rejection is based on a judicially created doctrine

grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or

improper timewise extension of the "right to exclude" granted by a patent and to prevent possible

harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed.

Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686

F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA

1970);and, *In re Thorington,* 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

        A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to

overcome an actual or provisional rejection based on a nonstatutory double patenting ground

provided the conflicting application or patent is shown to be commonly owned with this

application. See 37 CFR 1.130(b).

        Effective January 1, 1994, a registered attorney or agent of record may sign a terminal

disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37

CFR 3.73(b).

3.   Claims 1, 32 are provisionally rejected under the judicially created doctrine of

obviousness-type double patenting as being unpatentable over claims 6, 21, 25 of copending

Application No. 10/675777 (hereinafter '777).

        Although the conflicting claims are not identical, they are not patentably distinct from

each other because of the following observations.

        As per instant claims 1, 32, '777 claims 6, 21, 25 also recite determining for a instruction

during execution for a association of an indicator associated with receiving a bundle or

instruction in a instruction cache; associating a counter based on such determination and

incrementing a counter in response to the indicator association with the instruction or event

associated with the indicator. The event counting and instruction cache as recited by '777 are

construed as obvious representation to a runtime indicator (leading to a counter increment, in

which incrementing is count of number of instructions execution) and sending from the cached

instruction for execution of the instant claims. Further, '777 does not recite 'spare bit' for

indicator identifying whether the instruction is to be monitored, but based on the indicator

received into a cache, a bit type implementation as a slot within an received instruction would

have been a obvious feature, enabling a monitoring to receive the instruction and increment a

counter based on '777 paradigm.

4.   Claims 1, 32 are provisionally rejected under the judicially created doctrine of

obviousness-type double patenting as being unpatentable over claims 2, 10, 20 of copending

Application No. 10/675778 (hereinafter '778). Although the conflicting claims are not identical,

they are not patentably distinct from each other because of the following observations.

As per claims 1, 32, '778 claims 2, 10, 20 recite receiving a instruction with an indicator generated from a instruction cache, wherein upon determining that an indicator is associated with an instruction and a signal from the cache instruction, incrementing the counter each time the instructions is executed based on said cache signal. Even though '778 does not recite receiving bundle into a instruction cache and sending the received bundle for execution, this limitation of instruction associated with indicator from cache would made the sending a obvious step within runtime based on instruction being cached in view of the above association and counting event.

Further, '778 does not recite 'spare bit' for indicator identifying whether the instruction is to be monitored, but based on the indicator received into a cache, a bit type implementation as a slot within an received instruction would have been a obvious feature, enabling a monitoring to receive the instruction and increment a counter based on '778 paradigm.

5.   Claims 1, 32 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 3, 17 of copending Application No. 10/675872 (hereinafter '872).

As per instant claims 1, 32, '872 claims 3, 17 also recite instruction to be monitored and sent from cache instruction, determining whether an instruction in execution is related with an runtime range 'indicator'; and counting each event associated with the instruction if the instruction is associated with that range indicator. Even though '872 does not recite receiving bundle into a instruction cache and sending the received bundle for execution, said limitation of instruction with associated indicator would made the instruction cache receiving and sending obvious steps within runtime based on instruction being cached in view of the above known concept, and the incrementing responsive to association of instruction with the indicator. Even

though '872 explicitly recites that the indicator is a location within contiguous range, this
location-within- range limitation would be a obvious representation of any runtime indicator that
would characterizes as an event deemed for the counter to be incremented (in which
incrementing in terms of count of number of instructions execution) in view of the above
association determination. Further, '872 does not recite 'spare bit' for indicator identifying
whether the instruction is to be monitored, but based on the indicator received into a cache, a bit
type implementation as a slot within an received instruction would have been a obvious feature,
enabling a monitoring to receive the instruction and increment a counter based on '872 paradigm

6.   Claims 6, 34, are provisionally rejected under the judicially created doctrine of
obviousness-type double patenting as being unpatentable over claims 4, 12, 20 of copending
Application No. 10/675721 (hereinafter '721).

        As per instant claims 6, 34, 43, '721 claims 4, 12, 20 also recite determining for a
instruction during execution for a association of a indicator, shadow memory (Note: even
though '721 does not recite counter in shadow memory per se, a set of indicators being sent for
monitoring would have made the counter as obviously in the shadow memory); incrementing a
counter in response to the indicator association with the instruction, and responsive to which,
executing while incrementing said executing. At the time the invention was made, expediting
execution using instruction cache associated with profiling was known concept. Even though
'721 does not recite receiving bundle into a instruction cache and sending the received bundle for
execution, said limitation of instruction with associated indicator would made the instruction
cache receiving and sending obvious steps within runtime based on instruction being cached in
view of the above known concept, and the incrementing responsive to association of instruction

with the indicator. The instruction in the *routine of interest* as recited by '721 is construed as

obvious representation to a runtime instruction that requires some action (e.g. to monitor or to

trace/modify leading to a counter increment in which incrementing is in terms of count of

number of executions) of the instant claims. Further, '721 does not recite 'spare bit' for indicator

identifying whether the instruction is to be monitored, but based on the indicator received into a

cache, a bit type implementation as a slot within an received instruction would have been a

obvious feature, enabling a monitoring to receive the instruction and increment a counter based

on '721 paradigm

7.   Claims 1, 32 are provisionally rejected under the judicially created doctrine of

obviousness-type double patenting as being unpatentable over claims 1, 12, 23 of copending

Application No. 10/682385 (hereinafter '385).

        As per instant claims 1, 32, '385 claims 1, 12, 23 also recite executing instructions and

detecting indicators that specify counting of events associated with the executing (Note: even

though '385 recites data values in memory specifying counting event, a runtime event such as

those memory indicators can be analogous to on runtime indicator of the instant claim); and

counting each event associated with indicators. At the time the invention was made, expediting

execution using instruction cache associated with profiling was known concept. Even though

'385 does not recite receiving bundle into a instruction cache and sending the received bundle for

execution, said limitation of executing instructions associated with indicators would made the

instruction cache reception and the sending obvious steps within runtime based on instruction

being cached in view of the above known concept, and the incrementing responsive to

association of instruction with the indicator. Even though '385 explicitly recites that counting

events associated with execution based on detection of value indicators, this limitation would be

a obvious representation of any runtime indicator that would characterizes as an event deemed

for the counter to be incremented (in which incrementing is in terms of count of number of

executions) in view of the above association determination. Further, '385 does not recite 'spare

bit' for indicator identifying whether the instruction is to be monitored, but based on the

indicator received into a cache, a bit type implementation as a slot within an received instruction

would have been a obvious feature, enabling a monitoring to receive the instruction and

increment a counter based on '385 paradigm

Specification: Objection

8.   A preliminary examination of this application reveals that it includes terminology which

is so different from that which is generally accepted in the art to which this invention pertains

that a proper search of the prior art cannot be made. For example: "instruction cache unit" and

"data cache unit".

For examples, see following:

(i) "Instruction cache unit 300 <u>processes</u> instructions for execution ... instruction cache

unit 300 <u>determines</u> which instructions are associated with indicators … when instruction cache

unit 300 <u>determines</u> that an instruction …is present" (Specifications: 1st para, 2nd para, pg 23).

The actions termed as 'processes' and 'determines' (as underlined) used in a context where

instruction cache unit 300 is actually performing *processing* and *determining* amount to a

terminology not well-accepted in the closest and related computer fields of using *instruction

cache*. Cache is hardware storage means created as a specialized part from memory to store

frequently used data, not to process instructions; while cache controller would be a separate unit

in communication with cache to effectuate some operations upon the data contained or received

in cache. Without further hardware implementation in the Disclosure regarding how this cache

is equipped with some processing and determining capability being integral thereto, the above

language is deemed far-fetched and required correction.

(ii) '… signals for these instructions are <u>sent by</u> instruction cache unit 300 … cache unit

300, detects the indicators and <u>sends</u> signals to performance' (3rd para, pg. 23, 1st para pg. 24)

amounts to an action performed by instruction cache unit 300. This action of sending taken by a

cache cannot be construed as a terminology well-accepted by related arts. Code in execution can

send data to other parts of the computer or network. A dedicated controller in communication to

cache would reasonably be such hardware/software entity capable of *sending*, which is nowhere

disclosed to support the above hard-to-accept terminology. In other words, no standard cache is

well-known --without further teachings from the Specifications-- as being capable to actively

send data.

(iii) 'the data and indicators <u>are processed by</u> a data cache unit, such as data cache unit

216 … data cache unit <u>sends</u> signals indicating that marked memory locations' (last para pg. 24).

One familiar with the art of using or implementing cache cannot accept functionality of cache in

terms being able to *process* data and actively *transmit* signals, absent clear teaching provided by

the Disclosure to that regard.

Figure 31 of the Specifications gives a illustration of this 'instruction cache unit' and one

can see that registers or counter (as they included therein) contain instruction in register slots, but

these basic architectural structures cannot be equivalent of any form reminiscent of processor

capabilities (processing, sending, determining) as construed from the many parts of

Specifications.

9.   Applicant is required to provide a clarification of these matters or correlation with

art-accepted terminology so that a proper comparison with the prior art can be made. Applicant

should be careful not to introduce any **new matter** into the disclosure (i.e., matter which is not

supported by the disclosure as originally filed).

        When a terminology whose implication and usage is different from that which is

generally accepted in the art to which this invention pertains, applicant is required to provide a

clarification of these matters or correlation with art-accepted terminology so that a proper

comparison with the prior art can be made. Applicant should be careful not to introduce any new

matter into the disclosure (i.e., matter which is not supported by the disclosure as originally

filed). "Instruction cache" is a well-known concept and from there to disclosing and

implementing this "instruction cache" as though it possess processing capabilities, some

minimum requirement has to be set forth in the Specifications.

        To this effect, Applicants have to provide (emphasis added) on the onset (of the

Disclosure) a lexicographic redefinition to the 'instruction cache unit' and 'data cache unit'

within the very context of usage as described (i.e. a capability not to store cached data but rather

to process, determine and send) as per the amended Specifications, definition including at least a

minimal amount of features or functionalities for one to ascertain of the metes and bounds of this

redefined lexicography, i.e. bounds that **would not depart from** the original scope of the field of

endeavor and the explicit extent to which the Specifications (including Drawings) was originally

filed.

According the MPEP § 2111.01 (IV) the applicant is required to provide his own

lexicographer for term that is used in a sense that is not widely accepted in common usage.

> "… where an explicit definition is provided by the applicant for a term, that definition
> will control interpretation of the term as it is used in the claim. Toro Co. v. White
> Consolidated Industries Inc., 199 F.3d 1295, 1301, 53 USPQ2d 1065, 1069 (Fed.
> Cir. 1999) (meaning of words used in a claim is not construed in a "lexicographic
> vacuum, but in the context of the specification and drawings"). Any special meaning
> assigned to a term "must be sufficiently clear in the specification that any departure from
> common usage would be so understood by a person of experience in the field of the
> invention." Multiform Desiccants Inc. v. Medzam Ltd., 133 F.3d 1473, 1477, 45
> USPQ2d 1429, 1432 (Fed. Cir. 1998). See also Process Control Corp. v.
> HydReclaim Corp., 190 F.3d 1350, 1357, 52 USPQ2d 1029, 1033 (Fed. Cir. 1999)
> and MPEP § 2173.05(a). The specification should also be relied on for more than just
> explicit lexicography or clear disavowal of claim scope to determine the meaning of a
> claim term when applicant acts as his or her own lexicographer; the meaning of a
> particular claim term may be defined by implication, that is, according to the usage of the
> term in >the< context in the specification.

A shortened statutory period for reply to this action is set to expire ONE MONTH or

THIRTY DAYS, whichever is longer, from the mailing date of this letter.

For the sake of prosecution, the 'instruction cache unit' and 'data cache unit' as based on

the disclosure will be treated as 'instruction unit' and 'data unit' or post-fetching, decoding/pre-

execution stage while the term *cache* will be not be given weight (Note: this will be discussed in

the following 112 Rejection) such that weight to this term would be relinquished upon the

Applicant providing of a lexicographer regarding how 'cache' is structurally and functionally

implicated in the above units.

Claim Rejections - 35 USC § 112

10.   The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making
and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it
pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode
contemplated by the inventor of carrying out his invention.

11.   Claims 1, 6, 26, 29-30, 32, 34-35, 38-39, 49-53 are also rejected under 35 U.S.C. 112,

first paragraph. Specifically, since the claimed invention is not supported by either a specific and

substantial asserted utility or a well established utility for the reasons set forth above, one skilled

in the art clearly would not know how to use the claimed invention.

     Claim 1 recites 'receiving a bundle at an instruction cache unit … *determining by the*

instruction cache unit whether the bundle contains ... responsive to a determination … *sending*

by the instruction cache unit ... sending the bundle from the instruction cache unit'.

     Claim 32 recites 'responsive to receiving … determining *by the instruction cache unit …*

responsive to a determination that the bundle contains ... sending a signal *by the instruction*

cache unit to a performance monitor unit'.

     According to the analysis of the Specifications the "instruction cache unit" is nowhere

taught as explicitly destined to <u>cache instructions</u> but rather to operate as a processing entity

reminiscent of one equipped with programmatic or hardware functionality in terms of executed

actions of *processing*, *determining* then *sending*. In this light however, there no sufficient details

reminiscent of a processor element or software entity established inside this 'instruction cache

unit' in order for the 'instruction cache unit' 3106 (see Fig. 31 – registers 3108, counter 3110) to

determine, process and send as claimed. The mere fact of having some registers defining

instruction range (Specifications, bottom pg. 61 top pg. 62) along with a counter cannot readily

demonstrate that "instruction cache unit 3106" does have a processor capability, when in fact the

only processor disclosed (see processor 210, Fig 2; processor 102, Fig. 1) remains a large entity

encompassing instruction cache unit 214, which is necessarily not included inside instruction

cache unit 214 or cache unit 3106. According to well-accepted meaning for computer hardware

or architecture (emphasis added), registers or counter (see 3108, 3110 - Fig. 31 and related text)
cannot be equated to software/hardware entities capable of *determining*, *processing* received
data, and *configuring data for sending* down a stream in the absence of specific functional details
therefor; nor is it possible to equate register to a cache memory as registers are dedicated entities
per architecture whereas cache is a dedicated small portion of a larger memory.

**The 'instruction cache unit' as disclosed is not equipped with cache capability nor is
it equipped with processor functionality from the entire disclosure**. There is no explicit
description as to how instruction has been cached in the "instruction cache unit", i.e. an
established entity resident to the disclosed 'instruction cache unit' that actually maintain any
cached data, as the nomenclature entails. Nor are there sufficient processor capabilities (from the
entire Disclosure) established inside the 'instruction cache unit' to effectively corroborate to any
claimed functionality of 'instruction cache unit' as recited or even as illustrated in the
Drawings(see Fig. 31 and text).

Claims 1, 6, 26, 29-30, 32, 34-35, 38-39 are rejected for NOT providing a deliberate ,
fundamental and clear implementation detail regarding the 'instruction cache unit' in terms of
utility or capacity to cache instruction OR structural details corroborating that this 'instruction
cache unit' is equipped with processing capability.

Claims 49-53 also recites 'incrementing a counter inside an instruction cache unit'.
There is no processing capability within the disclosed 'instruction cache unit' and no caching
capability by the entity throughout the Disclosure and are rejected for non-enabling support. As
shown in Fig. 31, there is no sufficient structure in instruction cache unit **3106** that corroborates
to the fact that this very unit is actually equipped with software to increment a counter.

Performance monitoring unit is disclosed as tracking count numbers (Specs, pg. 63) and nowhere

in the Disclosure is there any teaching regarding an increment action upon counter **3110** which is

done by this very unit 3106. The 'instruction cache unit' is herein interpreted as a stage just prior

to execution of fetched and decoded instruction.

   The lack of actual "instruction caching" capability/utility or processor structure

deficiency will be treated as though <u>the host processor</u> connected to this entity (data or

instruction cache unit) <u>is the actual entity or device supporting the functionality</u> expressed by the

above misuse of language; that is, the phrase 'by the instruction cache unit' is given patentable

weight only to the extent of a broad form of instruction analyzing/processing capability (prior to

scheduling) for the sake of prosecuting the merits of the claims.

Claim Rejections - 35 USC § 103

12.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in
> section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are
> such that the subject matter as a whole would have been obvious at the time the invention was made to a person
> having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the
> manner in which the invention was made.

13.   Claims 1, 6, 26, 29-30, 32, 34-35, 38-39, 49-53 are rejected under 35 U.S.C. 103(a) as

being unpatentable over Gover et al, USPN: 5,752,062 (hereinafter Gover), and further in view

of APA (Admitted Prior Art: Specifications/Background: pg. 3-4).

   As per claim 1, Gover discloses a computer implemented method in an instruction cache

of a data processing system for monitoring execution of instructions, the method comprising:

   receiving a bundle at an instruction cache unit (sequencer unit dispatch unit - Fig. 1; col.

6 line 53 to col. 7 line 14 - Note: "instruction cache unit" treated as stage subsequent to fetching

– as from a cache - but prior to sending instruction to scheduler or execution unit -- see USC 112

Rejection), the bundle containing at least one instruction slot, wherein the instruction slot

contains an instruction (floating point, load/store - col. 4 line 44 to col. 5 line 30);

   responsive to receiving the bundle, determining by the instruction cache unit (col. 6 line

57 to col. 7 line 2- Note: *determining* treated as in conjunction with determining capability at the

pre-scheduler entity -- see USC 112 Rejection) whether the bundle contains an indicator of the at

least one instruction slot (e.g. exception field - Fig. 3 – Note: instruction in rename buffers

whose bits are analyzed/decoded reads on indicator of bundle received);

   sending the bundle from the instruction cache unit to a functional unit for execution of

the instruction (col. 5 lines 32-44; col 6 lines 59-62)

   Gover does not explicitly disclose that the indicator (of the received bundle) within at

least *one spare bit* identifies *the instruction as one that is to be monitored by a performance

monitor unit*; and responsive to a determination that the bundle contains the indicator within the

at least one instruction slot, sending a signal by the instruction cache unit (refer to USC 112

Rejection -Note: sending by instruction cache unit treated as a processing stage that send

instruction to subsequent stage, e.g. a performance monitoring unit) to a performance monitor

unit, *wherein upon receiving the signal, the performance monitor unit increments a counter* in

the instruction cache unit (Note: instruction cache unit treated as a stage prior to sending

instruction for execution) that is associated with the instruction, the incrementing providing a

count of a number of times the instruction is executed.

   Gover discloses special purpose registers used to store bit information hence has taught

spare bit used to store information other than the type of instruction itself (see Fig. 3). Further,

the indicators stored in said special registers are purported to enable performance monitoring unit

to take proper action (e.g. MMCRn, interrupt signal - col. 9 line 52 to col. 10 line 12; col. 8 line

42 to col 9 line 26) or to enable users to monitor events; such that an *interrupt* flag can be

decoded to trigger stopping of some monitoring or profiling action or such that preconfiguring of

bits can set condition under which counting can be enabled or disabled (e.g. *counts of accesses*

... instruction being executed - col. 10 line 40 to col 11 line 37), wherein profiling or counting

tracks number of events related to instruction such as access of variable or data, which is

analogous to well-known practices, as mentioned by APA . That is, APA teaches event-based

profiling and sample-based profiling, and according to the latter, *interruption* at some determined

intervals (see Specifications: pg. 3-4) enable recording events of interest. Based on the use of

some field or bits of special buffer to be decoded to indicate an exception or completion or a

cache miss (col 8 lined 9-12; col. 10 line 58-64) as taught by Gover's via action by instruction

dispatch unit, it would have been obvious for one skill in the art at the time the invention was

made to implement the flag in special register of the received bundle so that a bit or flag therein

indicate stopping of a monitoring as taught above, whereby the performance monitoring unit

upon receiving of such indicator, would start counting events or instruction execution (e.g.

incrementing a count of a number of times the instruction is executed) or stop if a exception or

interrupt (as in APA and Gover) due a predetermined threshold being exceeded based on Gover's

approach using fields and bit of special registers to track events.

      As per claim 6, Gover discloses wherein the counter is located in a shadow memory (col.

8, lines 26-39 – Note: special registers and PMCs with state or content – MMCRn -- maintained

via special privilege access mode and being kept in parallel with execution scheduling – see col.

11, lines 14-50 -- as informational support thereof, hence reads on shadowing type of

information kept in memory; see SSR col. 9 lines 36-57).

As per claim 26, Gover does not explicitly disclose comprising using a spare field in the

bundle to contain the indicator; but based on the indicators received by the execution unit sent by

the sequencer and arrangement of field/bit in rename buffer or special purpose registers to enable

decoding of information relevant to the instruction being fetched (col. 6 lines 58 to col. 7 line

25) based on which to enable some needed profiling by the performance monitoring unit (see

Fig. 6a-b; refer to claim 1), it would have been obvious for one skill in the art at the time the

invention was made to enable a special field in the bundle as set forth in claim 1 so that a spare

slot contains this indicator, among the other slots that are primarily allotted for the instruction per

se (see Fig. 3).

As per claim 29, Gover discloses (by virtue of the rationale in claim 1) responsive to a

determination that the bundle contains the indicator, beginning incrementing the counter,

wherein the counter tracks any subsequent instruction executed by an associated processor (e.g.

to correspond to a particular processor – col. 8, lines 46-55)

As per claim 30, Gover does not explicitly disclose

receiving a second bundle at the instruction cache; responsive to receiving the second

bundle, determining whether a second instruction in the second bundle contains a second

indicator; and responsive to a determination that the second bundle contains the second indicator,

ending the incrementing of the counter. But the scenario for receiving, determining responsive

to an indicator, and sending signal to a performance monitoring unit has been addressed in claim

1 to enable proper synchronization of scheduled instructions. Based on the fetching of

instructions from instruction unit 14 (see Fig. 2) for a dispatching/sequencer unit to store

instruction related information in special registers (GPR, FPRs col 5 lines 6 to col 6 line 6) and

how this stored information or bit/field indicators are used in Gover to signal interruption

whereby triggering/stopping profiling as by the Performance monitoring unit (Fig. 6A) in a

monitoring of processors by Gover (see claim 1), it would have been obvious for one skill in the

art at the time the invention was made to implement this receiving, determining of a monitoring

indication, and sending signal for the performance monitoring unit so that a second instruction

being fetched from the same dispatch unit, would be decoded in order for interpretation of the

indicator to trigger a signal to the performance monitory unit exactly as this has been

implemented for the first instruction from the fetched bundle as set forth in claim 1, because this

would enable thorough Gover' s multi-processor system synchronization of all fetched

instructions from cache as explained in the rationale of claim 1.

As per claim 32, Gover discloses computer program product comprising:

a computer recordable medium having computer useable program code for monitoring execution

of instructions, the computer program product comprising computer usable program code for:

receiving a bundle at an instruction cache unit, the bundle containing at least one

instruction slot, wherein the instruction slot contains an instruction (re claim 1);

responsive to receiving the bundle, determining by the instruction cache unit whether the

bundle contains an indicator within at least one spare bit of the at least one instruction slot (re

claim 1),

sending the bundle from the instruction cache unit to a functional unit for execution of

the instruction (re claim 1).

Gover does not explicitly disclose wherein the indicator identifies the instruction as one

that is to be monitored by a performance monitor unit, and responsive to a determination that the

bundle contains the indicator within the at least one instruction slot, sending a signal by the

instruction cache unit (refer to USC 112 Rejection -Note: sending by instruction cache unit

treated as a processing stage that send instruction to subsequent stage, e.g. a performance

monitoring unit) to a performance monitor unit, wherein upon receiving the signal, the

performance monitor unit increments a counter in the instruction cache unit that is associated

with the instruction, wherein the incrementing provides a count of a number of times the

instruction is executed.

But this indicator to signal incrementing a count has been addressed in claim 1.

As per claim 34, refer to claim 6.

As per claims 35, 38, 39 refer to claims 26, 29, 30 respectively.

As per claim 49, Gover discloses a computer-implemented method of monitoring

software performance in a data processing system, the computer-implemented method

comprising:

detecting an indicator (Fig. 3; col. 9 line 52 to col. 10 line 12) associated with one of an

instruction and a memory location unit of a processor;

Gover does not explicitly disclose responsive to detecting the indicator, incrementing a

counter in an instruction cache unit (Note: instruction cache unit treated as a stage prior to

sending instruction for execution) that is associated with the indicator; and analyzing, in a

performance monitor unit, a value of the counter to determine a performance of the data

processing system.

But detecting in a pre-execution stage of an indicator as to signal a course of action (e.g. incrementing a counter by a performance monitoring unit by Gover in light of APA) taken by profiling stage to enable or stop a counting of events has been addressed in claim 1.

As per claims 50-51, Gover discloses wherein responsive to the indicator being associated with an instruction, the incrementing occurs each time an instruction is executed (refer to claim 29; col. 17 lines 40-49); the incrementing occurs each time the memory location unit is accessed (*counts of accesses ... instruction being executed* - col. 10 line 40 to col 11 line 37; col. 17 lines 9-19).

As per claims 52-53, Gover discloses generating an interrupt from an interrupt unit responsive to the value of the counter exceeding a threshold value (col 10 lines 42-53);

including a criteria for the counter; and generating an interrupt from an interrupt unit responsive to meeting the criteria (col. 10 line 64-66).

Response to Arguments

14.   Applicant's arguments filed 4/20/09 have been fully considered but they are not persuasive. Following are the Examiner's observation in regard thereto.

Objection to the Specifications:

(A)   Applicants have submitted that as amended, the described functionality and related Drawings regarding the objected to entities has been more clear as to obviate the previous objections. The elements provided from the Drawings and from the entire Specifications are deemed not provided with proper support as this objection has been maintained and further explained in the USC 112 Rejection.

USC 112 Rejection (1st and 2nd paragraphs):

(B)   Applicants have submitted that the denoted 'Instruction cache unit" is sufficient to clarify

that an utility is established to provide actions of processing, determining, and sending. The

above observation is not persuasive in light of the now current Grounds of rejection.

35 USC § 103 Rejection

(C)   Applicants have submitted that 'instruction cache unit' as required by the claim language

contains additional logic to process data as this is shown in the Disclosure (Fig 31) and this

would not be same as Gover's fetching from a common instruction cache. The argument is

deemed largely moot because the change to the claim has made it necessary to provide a

readjusted ground of rejection therefor (including interpretation by the Examiner in conjunction

with the USC 112 rejection).

        In all, including the subject matter of claims 49-53, the claims stand rejected as set forth

in the Office Action.

Conclusion

15.   **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time

policy as set forth in 37 CFR 1.136(a).

        A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after

the end of the THREE-MONTH shortened statutory period, then the shortened statutory period

will expire on the date the advisory action is mailed, and any extension fee pursuant to 37

CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the mailing

date of this final action.

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The

examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Lewis Bullock can be reached on (571)272-3759.

The fax phone number for the organization where this application or proceeding is

assigned is (571) 273-3735 (non-official correspondence - please consult Examiner before

using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-

272-3609.

Any inquiry of a general nature or relating to the status of this application should be

directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application
Information Retrieval (PAIR) system. Status information for published applications may be
obtained from either Private PAIR or Public PAIR. Status information for unpublished
applications is available through Private PAIR only. For more information about the PAIR
system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR
system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Tuan A Vu/

Primary Examiner, Art Unit 2193

June 24, 2009